Міністерство освіти і науки України

Національний технічний університет України

“Київський політехнічний інститут”

Кафедра ОТ

ЗВІТ

про виконання комп’ютерного практикуму № 3

з дисципліни

“Основи об'єктно-орієнтованого програмування”

Тема: ДИРЕКТИВИ ПРЕПРОЦЕСОРА С++

|  |  |  |
| --- | --- | --- |
| Прийняв: Головченко М.М. |  | Виконав: |
|  |  | студент 2-го курсу  гр. ІП-44 ФІОТ  Майоров І.М.  Варіант 8 |

Київ 2015

# **1 ЦЕЛЬ РАБОТЫ**

Цель работы – изучить основные концепции объектно-ориентированного программирования. Изучить особенности использования классов и объектов, а также особенности применения конструкторов и деструкторов.

## **2 АНАЛИТИЧЕСКИЕ ВЫКЛАДКИ**

Целью введения концепции классов в С++ является предоставление программисту средств создания новых типов, которые настолько же удобны в использовании, как и встроенные. Тип является конкретным представлением некоторой концепции. Например, встроенный тип float вместе с операциями +, -, \* и т.д. представляет собой конкретное воплощение математической концепции вещественного числа. Определение 1. Класс - разновидность абстрактного типа данных в объектно- ориентированном программировании, характеризуемый способом своего построения. Наряду с понятием «объекта» класс является ключевым понятием в ООП. Определение 2. Класс – это определяемый пользователем тип. Класс должен быть объявлен до того, как будет объявлена хотя бы одна переменная этого класса. Конкретный экземпляр класса называется объектом.

**3 АЛГОРИТМ**

**АЛГОРИТМ МЕНЮ:**

**ШАГ 0.** Подключение необходимых библиотек

**ШАГ 1. ПОКА (ввод != 'q')**

**ШАГ 1. 1** Создать объект класса Triangle.

**ШАГ 1. 2** Вывести информацию об объекте

**АЛГОРИТМ СОЗДАНИЯ ОБЪЕКТА:**

**ШАГ 0.** Подключение необходимых библиотек

**ШАГ 1. СОЗДАТЬ** вектордля трёх точек **lol.**

**ШАГ 2. ВВОД** трёх точек в вектор

**ШАГ 3. ВВОД** строки в переменную **s**

**ШАГ 4. ЕСЛИ** строка **s** пуста

вернуть **Triangle(lol[0], lol[1], lol[2])**

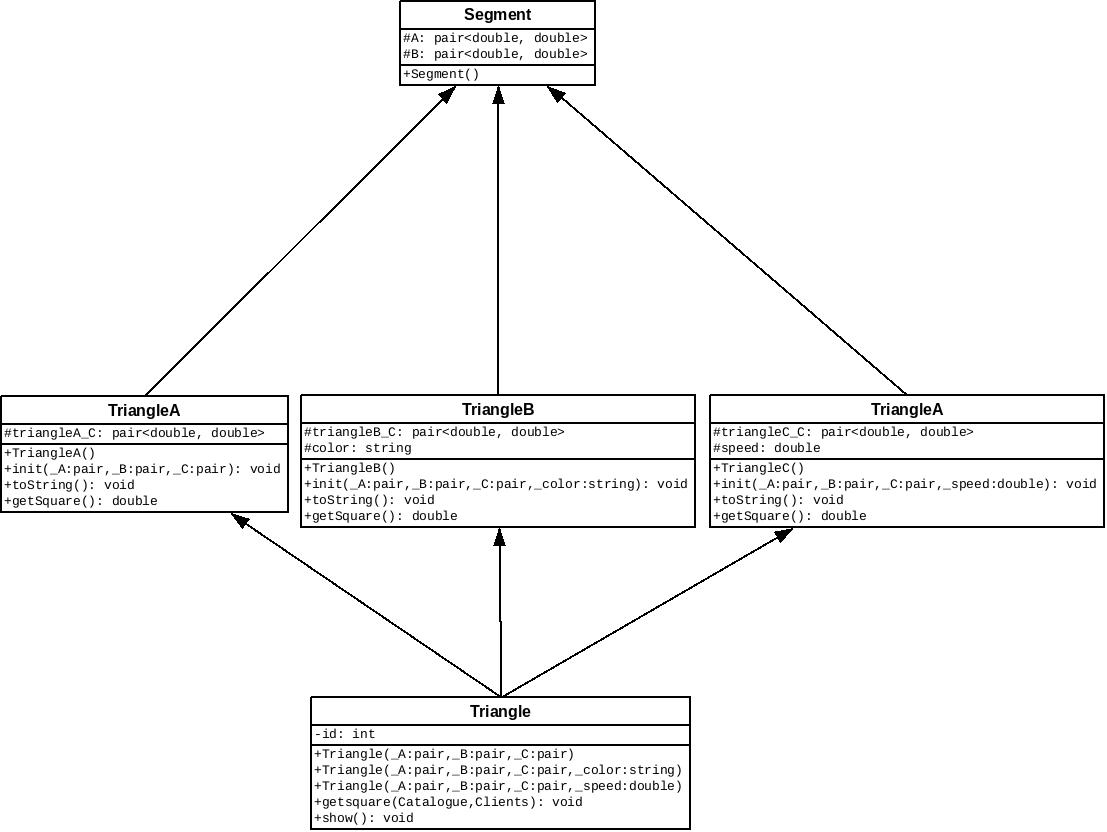
**ЕЩЕ ЕСЛИ s == число**

вернуть **Triangle(lol[0], lol[1], lol[2], число)**

**ЕЩЕ**

вернуть **Triangle(lol[0], lol[1], lol[2], s)**

**4 UML-ДИАГРАММА**



**5 КОД ПРОГРАММЫ**

#include "tools.h"

#include "segment.h"

#include "triangle.h"

using namespace std;

void eatline() {

while (getchar() != '\n')

continue;

}

bool is\_number(const string& s)

{

string::const\_iterator it = s.begin();

while (it != s.end()) {

if (isdigit(\*it) || (\*it) == '.')

it++;

else return false;

};

return true;

}

Triangle\* initialize() {

double x, y;

vector<pair<double, double> > lol(3);

for (int i = 0; i < 3; i++) {

cin >> x >> y;

lol[i] = make\_pair(x, y);

}

string s;

eatline();

getline(cin, s);

if (s.empty()) {

return new Triangle(lol[0], lol[1], lol[2]);

}

else if (is\_number(s)) {

double number = stod(s, NULL);

cout << "I'm here!" << endl;

return new Triangle(lol[0], lol[1], lol[2], number);

}

else return new Triangle(lol[0], lol[1], lol[2], s);

}

int main() {

puts("Igor Mayorov, IP 44, var. 8, level HARDEST.");

puts("Press any key to enter triangle or q to exit.");

while (getchar() != 'q') {

system("clear");

puts("Igor Mayorov, IP 44, var. 8, level HARDEST.");

puts("Press any key to enter triangle or q to exit.");

cout << "Enter 3 pairs of coordinates in following format: x y.\n"

"Then, double or string value." << endl;

Triangle \*lol = initialize();

cout << lol->getsquare() << endl;

}

return 0;

}

**6 ПРИМЕР РАБОТЫ ПРОГРАММЫ**

Данный скриншот показывает что программа работает правильно.

**7 ВЫВОДЫ**

В этой лабораторной работе я научился работать с классами. Узнал что такое инкапсуляция. Она предоставляет 3 способа доступа к данным, это public, protected и private. Научился работать с конструкторами по умолчанию, конструкторами инициализации и конструкторами копии. Еще в этой лабораторной работе использовал деструктор.